International Journal of Multiphysics
Volume 18, No. 3, 2024
ISSN: 1750-9548

Predicting Stress in Sanskrit Texts: A Deep Learning Approach to Sentiment Analysis

[1]Sabnam Kumari, [2]Amita Malik

[1]Computer Science & Engineering Department, Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Sonipat, Haryana, India, shabnam022@gmail.com

[2]Computer Science & Engineering Department, Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Sonipat, Haryana, India, amitamalik.cse@dcrustm.org

Abstract:
Sanskrit, one of the world's oldest languages, grammar plays major role in language translation, involving the structural arrangement of sentences through specific guidelines. Recently, there has been growing interest in the analysis of Sanskrit, particularly in translating the Bhagavad Gita into various languages. However, there is a lack of work validating the excellence of these English conversions. Advances in verbal models motorized by deep learning have not solitary facilitated conversions but also enhanced the sympathetic of languages and manuscripts through sentimentality examination. Despite these advancements, natural language processing (NLP) tasks such as machine conversion and sentimentality examination for Sanskrit have not been fully discovered, largely due to the scarcity of available data. To address these challenges, we present a sentiment analysis to predict the stress of Sanskrit texts using deep learning technique. We first perform the text preprocessing with the help of NLP transformer that considers word sequences to ascertain the accurate interpretation of words. Next, the XLNet based feature extraction is used to extracts meaningful features from the preprocessed results. We design the modified hyper spherical searching (MHSS) algorithm is used to selects the optimal features to reduce the dimensions. A dynamic dual-layer Q-learning (DDQL) model is present for sentiment analysis to classify the sentiments and predict the stress form Sanskrit text. We authenticate the effectiveness of the planned perfect using selected chapters and verses from the Bhagavad Gita across different translations. The proposed framework has demonstrated superior performance compared to existing methods for translation and sentiment classification.

Keywords: Natural language programming, sentimental analysis, Sanskrit language, DDQL, Bhagavad Gita

1. Introduction

Sanskrit, an ancient Indian language, is crucial to human culture. Hindu mythology calls Sanskrit the language of the gods because of its Vedic roots [1]. The Vedas, Upanishads, Mahabharata, Ramayana, and other intellectual, literary, and scientific works were written on it. Scholars worldwide study Panini's Sanskrit grammar for its structure and technique, which has affected comparative and modern linguistics [2]. Sanskrit affected Southeast Asian and India's culture and religion in addition to its intellectual and scholarly relevance. Hindu, Buddhist, and Jain ceremonies use Sanskrit hymns, chants, and mantras, demonstrating the language's religious and ritual qualities [3] [4]. Machine learning (ML) particularly deep learning (DL) copies have improved natural language processing (NLP) in recent decades [5][6]. Traditional NLP used systems based on rules and statistical models, which required personal involvement and struggled to understand and generate human language. DL models learn from enormous data sets and capture complex linguistic patterns, transforming NLP [7]. Word embeddings,

sequence-to-sequence, and attention mechanisms improve NLP automated translation, sentiment analysis, and summarization.

The combination of NLP with other AI technologies like computer vision and speech recognition has enabled multimodal applications, extending NLP's promise in healthcare, education, and entertainment [8]. These advances are changing how we use technology and making human-computer interactions more natural. DL techniques have revolutionized sentiment analysis by improve textual data sentiment extraction accuracy and efficiency [9] [10]. Models based on transformers like BERT and GPT continue to change sentiment analysis. The models accurately capture dependence over time and contextual interactions by judging sentence words' importance using attention mechanisms. Pre-trained on vast quantities of text, these models can be optimized on sentiment analysis datasets. They excel in irony, sarcasm, and complex emotions, which are necessary for sentiment analysis [11]. Businesses and researchers uses customer sentiment, company image, and public opinion need DL models to manage large amounts of data and learn from numerous sources [12].

Sanskrit analysis of sentiment offers numerous techniques to study ancient literature, texts, and philosophy [13]. One of the oldest languages with a rich literary and spiritual corpus, Sanskrit illustrates past societies' cultural, spiritual, and intellectual history. Sentiment analysis lets classify these writings' emotions, moods, and tones. In epic stories like the Mahabharata and Ramayana [14], analysis of sentiment can reveal characters' emotions and intentions, improving understanding. Software for education and digital humanities can benefit from Sanskrit sentiment analysis [15][16]. Sentiment analysis helps engage students in Sanskrit texts' emotive dynamics and intellectual discussions. DL has altered NLP by increasing human language understanding, comprehension, and generation [17]. DL in NLP extracts features from raw data, eliminating feature engineering. Traditional NLP is tedious and limited by rules and handmade features [18] to increases language comprehension, making sentiment analysis, machine translation, and summary analysis more accurate and contextual. DL in NLP improves scalability and flexibility across languages and domains [19] [20]. The key contributions of proposed model are given as follows.

1. The text preprocessing uses an NLP transformer to handle the unique characteristics of Sanskrit. This involves tokenization, lemmatization, and context-aware sequence analysis to preserve the syntactic and semantic nuances of the text.
2. We utilize XLNet, a powerful transformer-based model known for its robust performance in capturing complex language patterns. XLNet generate high-dimensional feature vectors that represent the underlying sentiments and themes within the Sanskrit text.
3. The modified hyper spherical searching (MHSS) algorithm is used to selects the optimal features to reduce the dimensions which discover best optimal features among multiples.
4. Dynamic dual-layer Q-learning (DDQL) model dynamically adapts to the complexities of sentiment classification. In DDQL, first layer classifies the primary sentimentality (positive, negative, or neutral), and the second layer refines this classification to predict specific stress levels.
5. We select chapters and verses from the Bhagavad Gita, Sanskrit text, and apply proposed model to validate the effectiveness. Different translations of the text are analyzed to ensure comprehensive validation.

The structure of this work is organized as follows. The second section presents the review of literature on NLP and sentiment analysis. The third section discusses the working process of proposed methodology with text preprocessing using NLP, feature extraction using XLNet, feature optimization using MHSS, sentiment analysis and stress detection using DDQL. The fourth section depicts the results and comparative analysis of proposed and existing models for Sanskrit texts. Finally, the paper concludes in fifth section.

2. Related works

Recent studies highlight the integration of these advanced techniques to tackle the complexities of sentiment analysis in various languages, including underrepresented ones like Sanskrit. This section synthesizes these developments, setting the stage for the proposed model's innovative contributions.

2.1 State of art works

Das et al. 2024 [21] examined colonialism's consequences on Bengali populations and sentiment analysis bias. The researchers reviewed Bengali sentiment analysis apps on GitHub and PyPI, focusing on age, faith, and nationality.

Kumar et al. 2023 [22] introduced a zero-shot learning-based cross-lingual sentimentality analysis (CLSA) method for Sanskrit manuscript sentiment analysis. Sanskrit lacks labeled data, so the study uses English, which has lots of data, to enhance sentiment evaluation.

Khaparde et al. 2023 [23] addressed the difficulties of identifying Sanskrit characters, especially in Devanagari, which is compounded by script complexity, identical character shapes, and many symbols. An image pre-processing model using the improved butterfly optimization algorithm (I-BOA) improves image quality before character identification.

Prakash et al. 2024 [24] investigated the role of NLP in improving AI-machine interactions by means of sophisticated semantic analysis. Computer vision and NLP have both benefited greatly from DL techniques, which are now essential for autonomous semantic analysis made possible by massive amounts of phonetic data and increasingly powerful computers.

Qiao et al. 2024 [25] investigated DL in NLP for cross-lingual sentiment evaluation, a difficult task due to globalization. NLP and sentiment evaluation definitions and history are covered first. They discuss linguistic and cultural disparities, annotation of data issues, and cross-lingual data obstacles in sentiment analysis. Experimental results evaluate the efficacy of models, and the finishes with a discussion of DL benefits, drawbacks, and future directions.

Tejaswini et al. 2024 [26] targeted early depression diagnosis, a crucial issue given the increased global prevalence of mood disorders and its devastating effects, like self-harm and suicide. Thy model uses Fast text embedding to improve text model, CNNs to extract global characteristics, and LSTM networks to capture local dependencies.

Sruthi et al. 2024 [27] explored DL models for sentiment analysis, stressing their ability to read and analyze text. RNN, CNN, and transformer-based models are for sentiment subtleties and contextual information. They explore how pre-trained word embedding, transfer learning, normalization, and hyperparameter adjustment affect model performance.

Jain et al. 2024 [28] embedded knowledge into sentiment analysis datasets to solve sentiment classification problems. Web ontology-based text is used to represent dataset features in DL models word embedding layer. Markov model-based auto-feature encoder extracts features and hierarchical convolutional attention networks classify them. They performed well on both the Facebook and Twitter ontology datasets.

Molenaar et al. 2024 [29] used social media data to study food security public health issues utilizing NLP methods like sentiment evaluation and topic modeling. The study examined 38,070 Australian tweets from three years and concluded that food security attitude was good, with changes tied to events like COVID-19 lockdowns. The study found no correlation between sentiment and engagement across themes however negative tweets had higher engagement. The findings show that NLP can follow food security debates and that public health decision-making requires more data and professional interpretation.

Ahmed et al. 2024 [30] examined how sentiment analysis may identify and neutralize terrorist content on Twitter using 1.6 million tweets and 24,000 tweets. Data preprocessing includes part-of-speech tagging, SentiWordNet sentiment score, and domain-specific word handling..

2.2 Problem description

While our proposed framework for sentimentality analysis of Sanskrit texts using deep knowledge and NLP techniques shows promising results, several challenges and limitations are associated with this work. A major challenge is the lack of high-quality, annotated datasets for Sanskrit texts. The meaning of Sanskrit words heavily relies on context, complicating the models' ability to interpret and translate texts without contextual understanding. Preprocessing Sanskrit texts also presents challenges such as word segmentation, where segmenting words correctly due to sandhi rules is complex, and normalization, which involves normalizing spelling and grammar variations to effectively process the text [21]-[31]. Feature selection is another critical aspect with the high dimensionality of Sanskrit texts produces a large number of features due to their rich vocabulary and complex structure. Select relevant features avoiding overfitting [31] is significant challenge, even with the use of optimization algorithm. Sentiment analysis specific challenges include the ambiguity and subjectivity of sentiments in texts, especially ancient and philosophical texts like the Bhagavad Gita, make accurate labeling and classification difficult. Validation also poses issues, as the model's effectiveness remains to be seen on other Sanskrit texts or translations beyond the selected chapters and verses from the Bhagavad Gita used for validation. There is risk of overfitting due to the limited amount of high-quality labeled data, reduce its effectiveness on unseen data. Lastly, the black-box nature of DL models, particularly those used for NLP, limits the interpretability and explainability of the model’s predictions. This lack of transparency makes it difficult to understand and trust the model’s outputs and analyze and interpret errors to iteratively improve the model. To address these challenges, future work can focus on expanding datasets through collaborative efforts, developing advanced preprocessing methods, exploring more efficient feature selection, testing the model on broader range of Sanskrit texts enhance the model interpretability predictions, and combining DL models.

3. Methodology

Fig. 1 provides a comprehensive overview of the methodology used to predict stress levels in Sanskrit texts through sentiment analysis. Our approach begins with utilizing a dataset from the Bhagavad Gita, which offers an extensive collection of Sanskrit language data rich in philosophical and emotional content. The first step in the process is text preprocessing, where the text undergoes several steps to prepare it for analysis. Initially, tokenization divides text into individual tokens, such as arguments or phrases, to facilitate computational manipulation. Normalization follows, standardizing the text format by converting all characters to lowercase, which helps reduce complexity and ensure consistency. Stop word extraction removes common words that do not add significant value to the analysis. Stemming reduces words to their root form, for example, "run" - "run", which helps combine the different uses of a word into a single unit. Finally, part-of-speech (POS) tagging assigns linguistic tags such as noun, verb, and adjective to each word to understand the syntactic structure of the text. Feature extraction is performed using XLNET, which is used to capture complex context and semantic relationships in text. XLNet uses a bidirectional environment to create a rich and comprehensive representation of features that includes complex meanings and relationships between terms. We use a modified higher spherical search (MHSS) algorithm to further refine the extracted features. Feature selection is improved by exploring a more spherical space and identifying relevant features that contribute significantly to emotion classification. The MHSS algorithm ensures that the feature set is compact and representative, further improving the efficiency and effectiveness of the analysis.

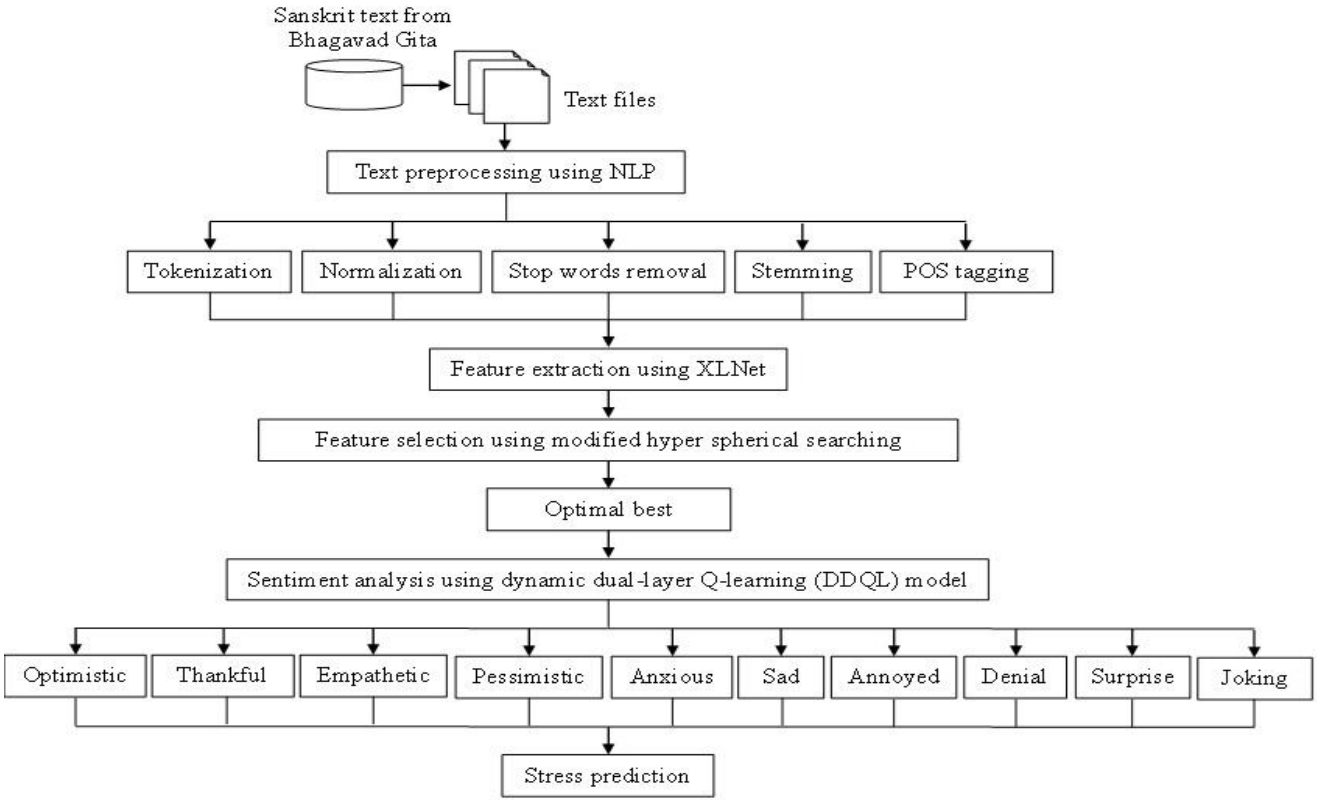


Fig. 1 Prediction of stress levels in Sanskrit texts using sentiment analysis

Sentiment analysis is performed using a dynamic dual-layer Q-learning (DDQL) model based on robustness learning, which dynamically adjusts the learning strategy based on input data and feedback. It divides emotions into different categories such as hope, gratitude, sympathy, pessimism, anxiety, sadness, anger, denial, surprise, and humor. Using two-stage Q-learning, the model adapts and refines its predictions, improving accuracy and robustness in sentiment classification. Emotion classifications derived from the DDQL model are analyzed to determine the level of stress reported in Sanskrit texts. This includes integrating emotional scores and identifying patterns that indicate stress. By studying the distribution and intensity of different emotions, we can infer the overall emotional tone and stress level in the text.

3.1 Text preprocessing

Text preprocessing is an important initial step in our sentiment analysis method, where we use NLP techniques [32] to convert raw Sanskrit text into a structured form suitable for analysis. We first pre-process the text with the help of NLP Converter, which finds the correct interpretation of the word order in their context.

- Tokenization: Text is divided into individual characters, which can be words or phrases. Tokenization helps extract meaningful units of text, making it easier to manage and analyze large amounts of text data [33] [34].
- Stop words removal: Frequently used arguments that do not add meaningful meaning to the text such as "and", "a", "is", and "in" are highlighted [35] [36]. Removing stop words helps reduce noise and improve parsing performance.
- Advanced contextual analysis: The NLP transformer goes beyond simple preprocessing. The transformer model captures these contextual dependencies and provides an accurate representation of the meaning of the text [37].

By applying these pre-processing steps, we ensure that the text data is clean, consistent, and well-structured, which provides a strong foundation for subsequent feature extraction and sentiment analysis processes. This

comprehensive pre-processing helps capture the rich linguistic and contextual nuances of Sanskrit texts, essential for accurate emotion classification and stress level prediction.

3.2 Feature extraction

Feature extraction is the process of transforming raw textual data into a set of landscapes that can be used for input into ML/DL models [38]. Features are variables or attributes that represent data, and classically mean patterns and relationships within text. Feature extraction in NLP involves extracting various linguistic and contextual features of text, such as word frequencies, syntactic structures, semantic meanings, etc. XLNet [39] is a state-of-the-art transformer model that excels at capturing complex context and semantic relationships in text. Preprocessed text is first tokenized into subword units using XLNet's tokenizer. The tokenized text is passed to the XLNET embedding layer, which converts the tokens into dense vectors. These vectors capture the semantic values of the tokens. The embeddings are processed by several transformer layers. These layers use self-focusing mechanisms to capture contextual relationships between tokens. Each layer modifies the representation at different levels by considering interactions between tokens. The output of the final transformer layer contains the context embeddings for each token. The final step is to combine the contextual embeddings to create a single feature vector representing the entire text [40].

3.3 Feature selection

Feature selection is the process of reducing the number of landscapes, which helps simplify the model and makes it computationally efficient and fast to train. By selecting only the most relevant features, the model focuses on the most important information, often leading to better performance and generalization. A model with fewer important features is easier to explain and understand. A modified hyper spherical searching (MHSS) algorithm is designed to efficiently select optimal features from a set of extracted features. It works by exploring the hyperspherical space to find the most relevant features, reducing the dimensions while retaining the key information needed for accurate sentiment analysis[41][42].The proposed evolution procedure is based on the study of the interior space of the hypersphere formed by the hyperspherical core and its essential atoms. MHSS algorithm is predictable to converge to single hyperspherical core whose constituent particles are uniformly spaced and have the same function value as the hyperspherical core.

$$Min\{F(p) \mid p \in P\} \tag{1}$$

$$subjectto\ j(p) \le and\, i(p) = 0 \tag{2}$$

The use of MHSS algorithm with cost function of different criteria shows effectiveness in a wide variety of optimization errands. The impartial of this optimization procedure is to minimalize the impartial function signified as F(p). The objective function (OF) influences set of choice variables restricted to a certain range of values defined by the P variable.

$$P_{h,Min \le P_h} \le P_{h,Max} \tag{3}$$

Factors measured in this study are initial people size B_{pop} and amount of HSCs. The optimization process starts by randomly generate $[P_{h,Min}, P_{h,\max}]$ population of persons. Also, each adjustable xi is randomly designated from a constant probability delivery. In the background of B-dimensional optimization, subdivision is signified by means of 1 × B type vector, $[x1, x2, ...xB]$. p_h for h= 1, · · ·,B are having these decision variable star. In the MHSS algorithm, a particle cycle is formed in a population of N particles. The remaining particles are assigned by seeing the supremacy of threshold decision (TD), which are inversely comparative to the objective function value (OFV). Therefore, $ofd_{TD} = F_{TD} - Max_{TDt}\{F\}$ the regularized supremacy at each TD is distinct as follows.

$$C_{TD} = \left| \frac{ofd_{TD}}{\sum_{h=1}^{nsc} ofd_h} \right| k_\alpha, k_v \tag{4}$$

Therefore, the first number of these particles in a TD is randomly selected for each TD from among all remaining particles $\{C_{TD} \times (B_{pop} - B_{TD})\}$. Given an n-dimensional hypersphere, the value of 'R' between [$R_{\min}, R_{Max}$] is defined as follows.

$$R^2 = \sum_{h=1}^{B} (X_{h,center} - X_{h,particle})^2 \tag{5}$$

Upon the alteration and assessment of the variable quantity θ_T and R the search course of the subdivision arrives its achievement phase. The people engage in the development of looking for HSs, taking into version together the TD and the operatives that are quantified by $Y[R_{\min}, R_{Max}, X_{R_{Angle}} \& TD]$. During this period, the TD merges with its corresponding counterpart and forms particle cluster. We define the OFV of each set as follows.

$$sof = fsc + \gamma Mean\{F_{Particles\ of\ TD}\} \tag{6}$$

Decreasing the value of TD affects the determination of semi objective function (sof), while increasing this value affects the position of the particle in sof detection. Also, a worth of 0.1 was used for the mutable 'c'. The fake particle detection process is simulated by selecting several fake particles with high 'sof' in MHSS and combining them with other TDs. Also, the difference between each set can be efficiently handled using the equation below.

$$dsof = sof - Maxgroups\{sof\ of\ groups\} \tag{7}$$

The mean average peak point (ap) of each TD is compute as follows.

$$ap = \left| \frac{ntof}{\sum_{h=1}^{nsc} ntof_h} \right| \tag{8}$$

$$ap = [ap_1, ap_2, ...ap_{Btd}] \tag{9}$$

Furthermore, this specific phenomenon smears to the growing subset of all TDs founded on APs. It undergoes a transformation by acquiring new particles and then becomes a new TD using special technique. The algorithm 1 describes the working process of feature selection using the MHSS.

Algorithm 1 Feature selection using MHSS

Input: Number of features, maximum iteration and threshold condition
Output: Best optimal features

1. Begin;
2. Define the function associated with the optimization tasks
$$Min\{F(p) \mid p \in P\}$$
3. Fix certain range of values form the set P variable. $P_{h,Min \le P_h} \le P_{h,Max}$
4. Compute normalized dominance at each TD $C_{TD} = \left| \frac{ofd_{TD}}{\sum_{h=1}^{nsc} ofd_h} \right| k_\alpha, k_v$
5. Define objective function value (OFV) of each set
$$sof = fsc + \gamma Mean\{F_{Particles\ of\ TD}\}$$
6 End For
7. Compute difference between each set
$$dsof = sof - Maxgroups\{sof\ of\ groups\}$$

8. Compute ap of each TD $ap = \left| \frac{ntof}{\sum_{h=1}^{nsc} ntof_h} \right|$
9. End for
10. End While
11. Find the best position and fitness value

3.4 Sentimental analysis and stress prediction

For sentiment analysis and stress prediction in Sanskrit texts, we employ a dynamic dual-layer Q-learning (DDQL) model. The DDQL model incorporates two layers of Q-learning agents. The first layer focuses on coarse sentiment classification, while the second layer refines these classifications for more detailed sentiment analysis. The first layer agent processes the extracted features (from XLNet) and assigns preliminary sentiment labels, such as positive, negative, and neutral. This layer uses a standard Q-learning mechanism [43] where the agent explores various actions and updates its Q-values based on the rewards received. A second-layer agent takes raw emotion labels as input and organizes them into specific emotion categories, including hope, gratitude, empathy, pessimism, anxiety, sadness, annoyance, denial, surprise, and humor. This layer uses a sophisticated Q-learning process that takes into account nuances and nuances of text and provides detailed sentiment analysis. The reward system in the DDQL is designed to provide positive rewards for correct sensory categorizations and penalties for incorrect ones. Additionally, the reward structure can be adjusted to prioritize certain sensations associated with predicting stress. After classifying the sentiments, the DDQL model combines the sentiment scores to predict the overall level of stress expressed in the text. The national space (T^1) is extracted at Δs = 1 h. SOC and s represent the state of custody and period stage of the battery, correspondingly. It is significant to note that generation and load material is indirectly comprised in the period steps, as the cohort and load data outlines are time-adjusted during offline optimization.

$$t_s^1 = [soc, s] \in t \tag{10}$$

SOC is circumscribed by extreme and minutes bounds as follows.

$$soc_{Min} \le soc(s) \le soc_{Max} \tag{11}$$

The state space is discredited as follows.

$$T_{discrete}^1 = \{T_{h,g}\} \tag{12}$$

At apiece period step s (1 h), one deed is designated from the achievement space (Mf):

$$Mf = \{m \mid 0, \pm 10\%, \pm 20\%, \pm 30\% ..... \pm 100\%\} \tag{13}$$

The symbol (-) means charging, (+) means discharging and 0 means inactive. The percentages depend on the maximum capacity of the battery depending on the battery and its inverter rating.

$$X_s^{batt} = X_{Max}^{batt} \times Mf \tag{14}$$

Q-learning is used to minimize control smuggled from the network; thus, the recompense purpose as follows.

$$R^1(t_s, m_s) = -X_s^{Grid} \times \Delta s \times tariff - D \tag{15}$$

where X_s^{Grid} is the smuggled grid power and optimizes as follows.

$$X_s^{Grid} = E_s^{net} - X_s^{batt} \tag{16}$$

where $E_s^{net} = E_s^{demand} - E_s^{net}$ (net call).

By updating the Q-table, the manager tries each choice once and selects the one with the upper most upcoming recompense until the agent learns to maximize the worth of the state-action pair. Adaptive and greedy functions correspond to exploration (ε) and exploitation, respectively. Algorithm 2 describes the working process of sentimental analysis and stress prediction using DDQL.

Algorithm 2 Sentimental analysis and stress prediction using DDQL

Input: Best features, initial Q-values, coarse sentiment labels
Output: Sentimental analysis and stress detection

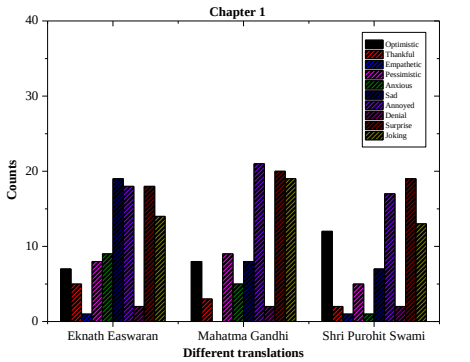
1. Begin;
2. Initialize the population
3. Define the load data profiles of time-adjusted during offline optimization. $t_z^1 = [soc, s] \in t$
4. Define the state space $T_{discrete}^1 = \{T_{h,g}\}$
5. While do
6. Compute smuggled grid power $X_z^{Grid} = E_z^{net} - X_z^{batt}$
7. Find normalised alteration between calculated and real solution $C_{F-R} = \left(E_z^{Net(real)} - E_z^{Net(forecast)}\right) / E_z^{Net(real)}$
8. Define sum of immediate rewards and future discounted rewards $r_z^{\pi} = r(t_z, m_z) + \sum_{h=1}^{\infty} \gamma^h . R(t_{z+h}, m_{z+h})$
7. End while
8. Compute the activity-value function by iterative updating $Y(t_z, m_z)$ through experience $Y(t_z, m_z) = Y(t_z, m_z) + \alpha[r(t_z, m_z) + \gamma \min Y(t_{z+1}, m_{z+1}) - Y(t_z, m_z)]$
9. Find the best output value
10. End

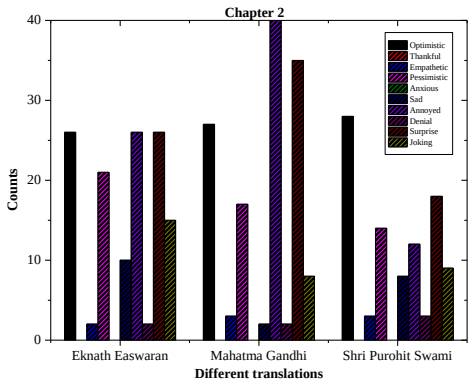
4. Results and Discussion

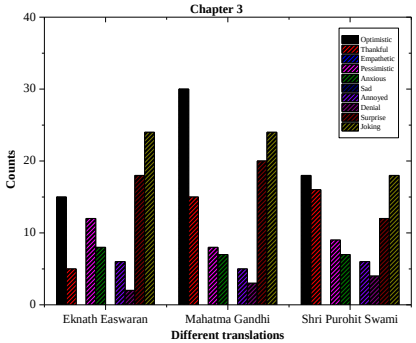
This section provides the results and comparative analysis of proposed and existing sentiment analysis and stress prediction models. We utilized the XLNet+DDQL model for analyzing sentiments and predicting stress in the Bhagavad Gita. The Bhagavad Gita, comprising 18 sections, landscapes dialogues among Lord Krishna and Arjuna on various topics, counting the philosophy of Karma, and is symbolically organized in alignment with the 18-day Mahabharata war. In Fig. 2. the sentiments in first chapter reveal a deep sense of inner conflict and emotional turmoil experienced by Arjuna on the battlefield. Mahatma Gandhi's translation displays a mix of pessimistic and annoyed sentiments, reflecting Arjuna's frustration and despair. Eknath Easwaran's version also highlights these sentiments, though with a slight emphasis on optimism, suggesting a more hopeful perspective amidst the struggle. In chapter 4, Krishna's clarification of the eternal nature of his wisdoms is met with awe and stimulus. Isvara's translation shows great hope and wonder, emphasizing the profound influence of Krishna's wisdom. Gandhi's translation combines hope, gratitude, and rage, suggesting a mixture of gratitude and despair at realizing this eternal truth. Swami's version is a balanced vision of hope and wonder, reflecting the enlightenment of the teachings.

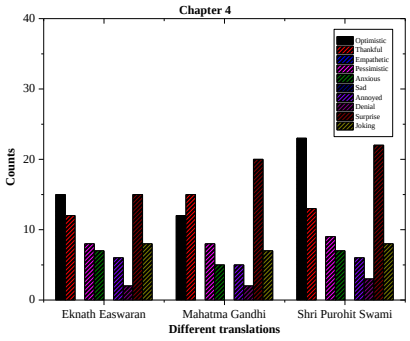
In chapter 5, The theme of finding pleasure in rejection is met with a variety of emotional responses. Ishvara's translation has sentiments of hope and gratitude, which lends a warm welcome to the idea of sacrifice. Gandhi's translation shows a mixture of hope and sadness, reflecting the challenge to reject attachment. Swami's version strikes a balance, reflecting hope and wonder, suggesting a nuanced understanding of the joys of renunciation. Chapter 6 discusses the importance of meditation, which is emphasized by significant positive emotions. There is a high level of faith and compassion in the interpretation of Ishvara, which reflects the beneficial effects of meditation. Chapter 7 analyzes Krishna's discourse on wisdom and how enlightenment evokes complex emotions. Isvara's translation shows a mixture of hope, compassion, and pessimism, indicating a multifaceted acceptance of these teachings. In chapter 8, According to Krishna, the eternal nature of the soul is well understood in the interpretation of God, which is characterized by feelings of faith and gratitude. Gandhi's translation shows a mixture of hope and sadness that reflects the struggle to understand eternity. Chapter 9 discusses the creative reception of bhakti teachings that translate Isvara as meaning hope and gratitude, indicating a welcoming response to the path of bhakti. Gandhi's translation shows faith and compassion, while Swami's version balances faith and compassion, reflecting a harmonious understanding of the royal path. In chapter 10, The description of Krishna's divine manifestations received a positive response. Chapter 11 presents Arjuna's experience of Krishna's universal form, while Chapter 12 depicts the path of devotion through love that evokes positive emotions in an interpretation of God filled with compassion and faith.

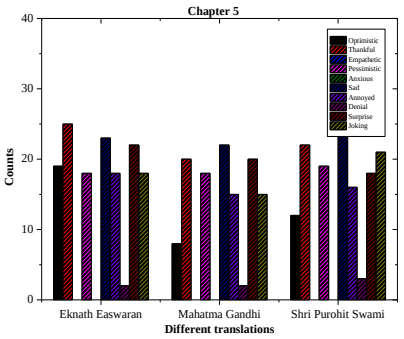
The conflict between body and soul creates a deep emotional response in chapter 13. Isvara's rendering expresses a high degree of compassion and sadness, indicating a deep understanding of this concept. Gandhi's translation reflects compassion and sadness, while Swami's version balances compassion and sadness, offering a thoughtful commentary on these teachings. Chapter 14 explains that there is a complex emotional response to describing the three gunas (modes of nature). There are significant feelings of surprise and excitement in Isvara's translation that reflect the complexity of Guna. Chapter 15 discusses the creative understanding of Krishna's interpretation of the Supreme Being translated by Isvara, which is characterized by faith and compassion. Chapter 16 illustrates the difference between divine and demonic ways that evoke significant emotional responses.

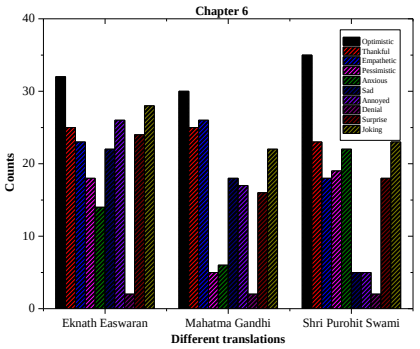


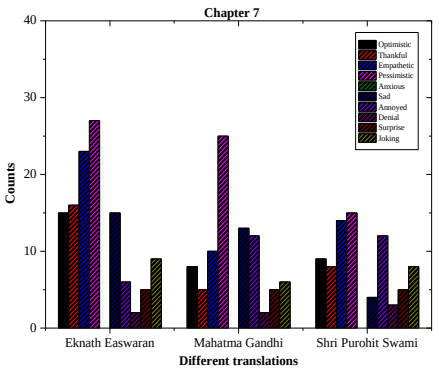


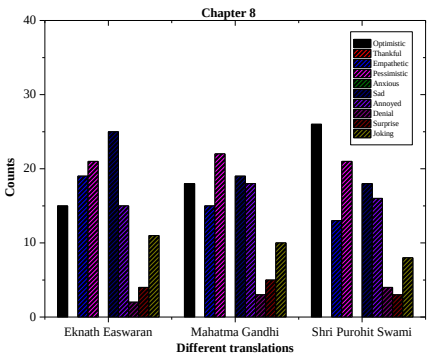




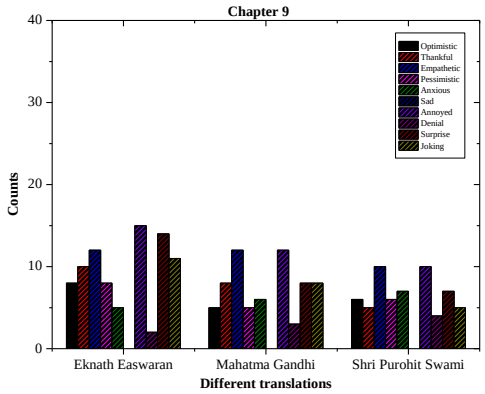


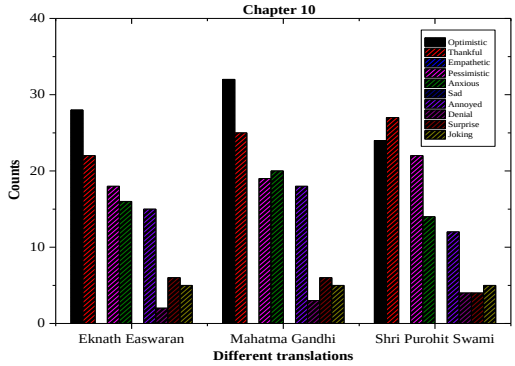


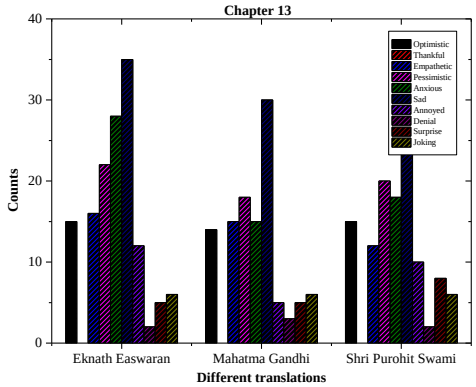


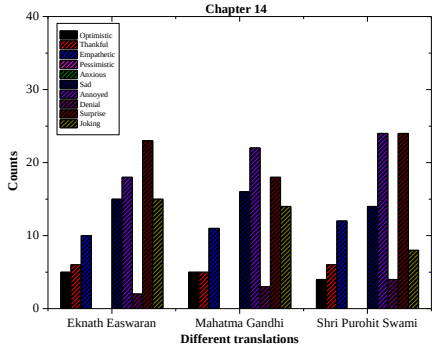


International Journal of Multiphysics
Volume 18, No. 3, 2024
ISSN: 1750-9548









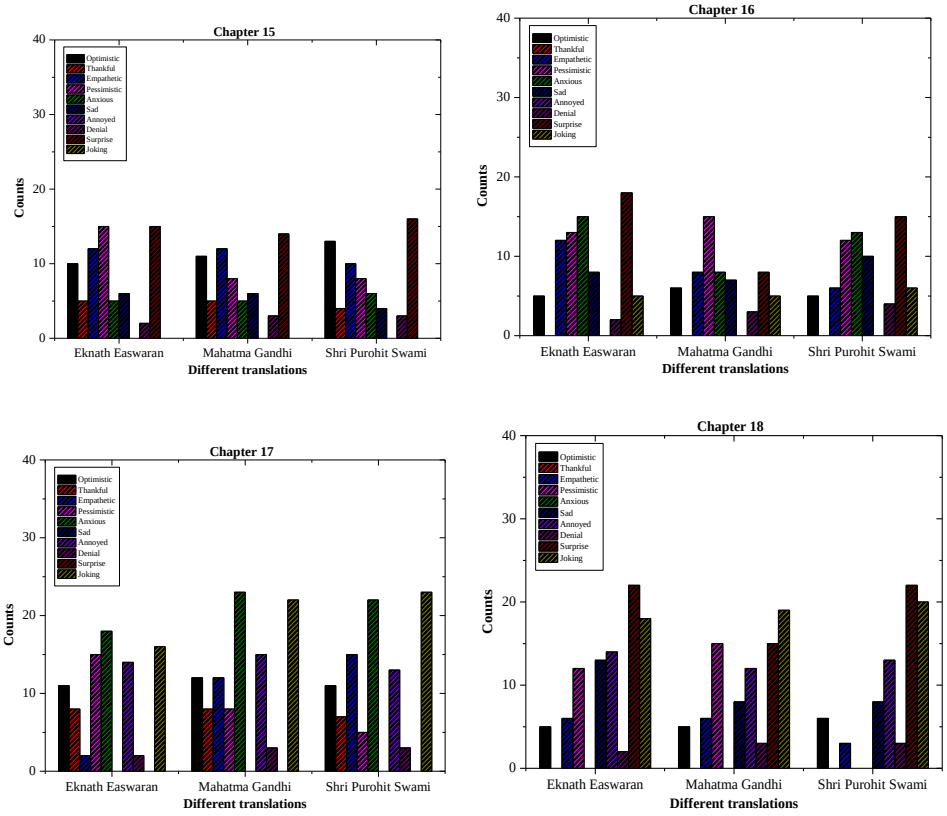


Fig. 2 Chapter-wise sentiment analysis, chapter 1- the war within, chapter 2 - self realization, chapter 3 - selfless service, chapter 4 - wisdom in action, chapter 5 - renounce and rejoice, chapter 6 - the practice of meditation, chapter 7 - wisdom and realization, chapter 8 - the eternal godhead, chapter 9 - the royal path, chapter 10 - divine splendor, chapter 11 - the cosmic vision, chapter 12 - the way of love, chapter 13 - the field and the knower, chapter 14 - the forces of evolution, chapter 15 - the supreme self, chapter 16 - two paths, chapter 17 - the power of faith and chapter 18 - freedom and renunciation.

The results presented in Table 1 show the Jaccard and Camus similarity scores for predicting emotions in different chapters of the Bhagavad Gita evaluated using the XLNet+ DDQL model. The data compares three translation pairs: Eknath-Mahatma, Mahatma-Purohit, and Purohit-Eknath. Analysis of overall trends revealed that the consistency of emotion predictions varied between episodes and translation pairs. For example, Chapter 10 shows particularly high similarity scores, especially for the Purohit-Eknath pair, where the Jacquard similarity score reaches 0.954 while the Dice similarity score is 0.894. This indicates a high degree of agreement between the translations of this chapter. In contrast, Chapter 13 shows low congruence scores, with the Purohit-Eknath pair showing a low dice congruence score of 0.680, indicating high variability in predicting emotions. Comparing the translation pairs, the Eknath-Mahatma pair shows relatively stable similarity scores, with an average Jaccard similarity score of 0.836 and an average Dice similarity score of 0.817. This consistency suggests relatively uniform agreement between the two translations across chapters. However, the Mahatma-Prohit pair has slightly lower mean scores with a mean Jaccard congruence of 0.818 and a mean Dice congruence of 0.825, reflecting moderate variation in emotion predictions,

Table 1 Results of predicted sentiments using XLNet+ DDQL model for selected pairs of translations form Bhagavad Gita

Chapter	Translations				
	Eknath-Mahatma	Mahatma-Purohit	Purohit-Eknath	Eknath-Mahatma	Mahatma-Puroh
	Jaccard similarity score			Dice similarity score	
Chapter-1	0.856	0.745	0.856	0.848	0.758
Chapter-2	0.878	0.758	0.845	0.862	0.858
Chapter-3	0.863	0.775	0.853	0.824	0.825
Chapter-4	0.758	0.771	0.862	0.885	0.806
Chapter-5	0.775	0.754	0.733	0.805	0.869
Chapter-6	0.776	0.805	0.834	0.786	0.867
Chapter-7	0.858	0.815	0.772	0.898	0.838
Chapter-8	0.825	0.836	0.846	0.759	0.841
Chapter-9	0.865	0.826	0.787	0.754	0.789
Chapter-10	0.842	0.898	0.954	0.862	0.878
Chapter-11	0.863	0.845	0.827	0.824	0.825
Chapter-12	0.875	0.748	0.898	0.848	0.788
Chapter-13	0.754	0.798	0.869	0.748	0.714
Chapter-14	0.866	0.848	0.830	0.766	0.763
Chapter-15	0.895	0.867	0.839	0.738	0.815
Chapter-16	0.798	0.888	0.978	0.825	0.868
Chapter-17	0.798	0.857	0.857	0.811	0.898
Chapter-18	0.896	0.898	0.900	0.868	0.847
Mean	0.836	0.818	0.852	0.817	0.825

The mean Jaccard similarity score for the Purohit-Eknath pair was 0.852 and the Dice similarity score was 0.832, indicating generally good agreement, but with some variation. Further inspection reveals that Chapters 10, 15, and 16 have high similarity scores, suggesting more stable mood predictions. On the other hand, chapters 13 and 14 show low similarity scores, which may lead to different sentiment interpretations due to more complex or nuanced passages. Average similarity scores across chapters indicate excellent overall agreement, with an average Jaccard similarity score of 0.836 and an average Dice similarity score of 0.825. These results highlight the model's generally useful performance in predicting sentiment, although they also point to areas that require further refinement.

Table 2 Performance comparison of proposed and existing models for predicted sentiments

Models	Values in %				
	Accuracy	Precision	Recall	Jaccard	Dice
	Eknath-Mahatma				
S-BERT [48]	64.409	60.878	59.578	0.485	0.337
MPNet [49]	69.665	66.134	64.834	0.508	0.393
XLNet [50]	74.921	71.390	70.090	0.425	0.449
Key-BERT [51]	80.177	76.646	75.346	0.456	0.505
BERT+DL [31]	85.433	81.902	80.602	0.526	0.561
XLNet+ DDQL	95.689	92.158	90.858	0.836	0.817
	Mahatma-Purohit				
S-BERT [48]	63.578	61.974	58.844	0.418	0.345
MPNet [49]	68.834	67.230	64.100	0.433	0.401
XLNet [50]	74.090	72.486	69.356	0.462	0.457
Key-BERT [51]	79.346	77.742	74.612	0.415	0.513
BERT+DL [31]	84.602	82.998	79.868	0.497	0.569
XLNet+ DDQL	94.858	93.254	90.124	0.818	0.825
	Purohit-Eknath				
S-BERT [48]	63.409	61.877	60.177	0.425	0.352
MPNet [49]	68.665	67.133	65.433	0.459	0.408
XLNet [50]	73.921	72.389	70.689	0.499	0.464
Key-BERT [51]	79.177	77.645	75.945	0.515	0.520
BERT+DL [31]	84.433	82.901	81.201	0.563	0.576
XLNet+ DDQL	94.689	93.157	91.457	0.852	0.832

The results for sentiment analysis on the Eknath-Mahatma translation pairs reveal significant variations in performance across different models, as shown in Fig. 3. Accuracy is highest with the XLNet+ DDQL model,

achieving an impressive 95.689%. It represents an improvement of 10.256% over BERT+DL, which has an accuracy of 85.433%.

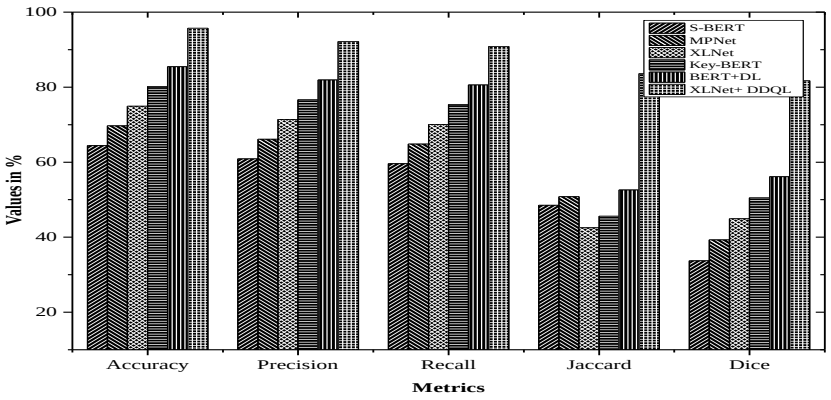


Fig. 3 Results of models for sentimental analysis on Eknath-Mahatma translation pairs

The accuracy increase from Key-BERT, with 80.177%, to XLNet+ DDQL is 15.512%, further highlighting the significant leap in performance. Precision also shows a marked improvement with the XLNet+ DDQL model, scoring 92.158%. This is 10.256% higher than BERT+DL, which has a precision of 81.902%. Precision increases by 25.280% from S-BERT, with 60.878%, to XLNet+ DDQL, indicating a substantial enhancement in correctly identify positive sentiment instances among the predictions. In terms of recall, XLNet+ DDQL leads with 90.858%, reflecting an increase of 10.256% from BERT+DL, which has a recall of 80.602%. This is an improvement from MPNet, with 64.834%, and S-BERT, with 59.578%, shows increases of 26.024% and 31.280%, respectively. The high recall of XLNet+ DDQL suggests a significant gain in detecting all relevant instances of positive sentiment. The Jaccard similarity score for XLNet+ DDQL stands at 83.6%, which is a considerable improvement of 31% over S-BERT, with 48.5%, and 31.8% over MPNet, with 50.8%. Finally, the Dice similarity score for XLNet+ DDQL reaches 81.7%, showing an improvement of 25.6% compared to S-BERT, which has score of 33.7%, and 42.4% increase from MPNet, with 39.3%. The improvement shows the effectiveness in capturing the overlap between predicted and actual sentiment instances. These improvements highlight its superior performance in sentiment analysis for the Eknath-Mahatma translation pairs, marking a significant enhancement over previous models.

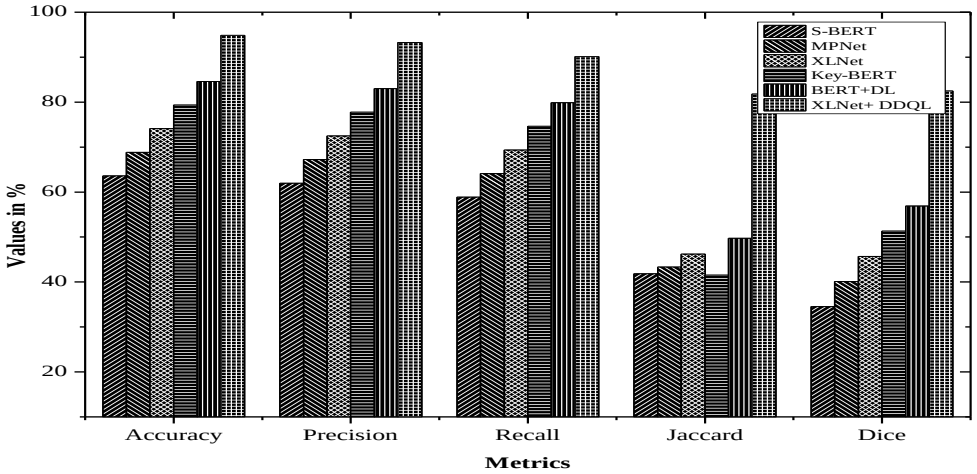


Fig. 4 Results of models for sentimental analysis on Mahatma-Purohittranslation pairs

The results for sentiment analysis on the Mahatma-Purohit translation pairs, as depicted in Fig. 4, reveal notable variations in model performance. Accuracy is highest with the XLNet+ DDQL model, achieving 94.858%. It represents a significant improvement of 10.256% over BERT+DL, which has an accuracy of 84.602%.

The increase in accuracy from Key-BERT, with 79.346%, to XLNet+ DDQL is 15.512%, shows substantial leap in performance. Precision also shows a marked enhancement with XLNet+ DDQL, scoring 93.254%. This is 10.256% higher than BERT+DL, which has a precision of 82.998%. Precision improves by 31.280% from S-BERT, with 61.974%, to XLNet+ DDQL, indicating a significant advancement in correctly identifying positive sentiment instances. In terms of recall, XLNet+ DDQL leads with 90.124%, reflecting an increase of 10.256% from BERT+DL, which has recall of 79.868%. It marks an improvement over MPNet, with 64.100%, and S-BERT, with 58.844%, shows increases of 26.024% and 31.280%, respectively.

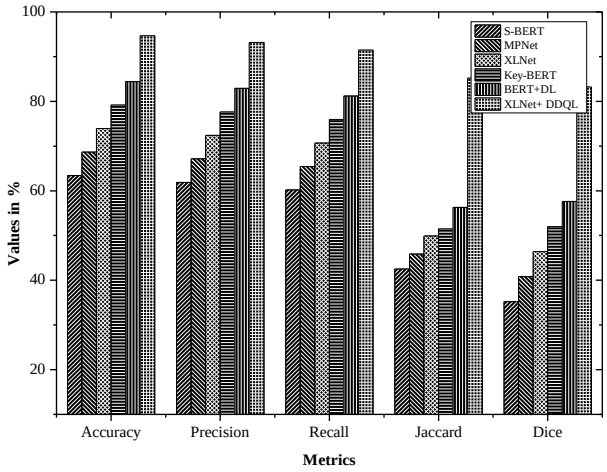


Fig. 5 Results of models for sentimental analysis on Purohit-Eknathtranslation pairs

The results for sentiment analysis on the Purohit-Eknath translation pairs, as shown in Fig. 5, highlight distinct performance variations across different models. Accuracy is highest with the XLNet+ DDQL model, achieving 94.689%. This marks a substantial improvement of 10.256% over BERT+DL, which has an accuracy of 84.433%. The accuracy increase from Key-BERT, with 79.177%, to XLNet+ DDQL is 15.512%, reflecting a significant leap in performance. Precision shows a similar trend with XLNet+ DDQL scoring 93.157%. This is 10.256% higher than BERT+DL, which has precision of 82.901%. Precision improves by 31.28% from S-BERT, which has 61.877%, to XLNet+ DDQL, indicating a substantial enhancement in correctly identifying positive sentiment instances. In terms of recall, XLNet+ DDQL leads with 91.457%, showing an increase of 10.256% over BERT+DL, which have recall of 81.201%. This represents an improvement from MPNet, with 65.433%, and S-BERT, with 60.177%, with increases of 26.024% and 31.28%, respectively. The high recall of XLNet+ DDQL suggests a significant gain in detecting all relevant instances of positive sentiment. The Jaccard similarity score for XLNet+ DDQL stands at 0.852, reflecting improvement of 40% over S-BERT, with 0.425, and 42.8% over MPNet, with 0.459. The results highlights the model's enhanced performance in terms of the amount of properly identified sentiment instances comparative to the total amount of predicted and actual instances. Dice similarity score for XLNet+DDQL reaches 0.832, shows improvement of 26% compared to S-BERT, which has a score of 0.352, and 42.4% increase from MPNet, with 0.408. Fig. 6 displays the keywords extracted from each chapter across various translations. To obtain a more detailed perspective, a focused analysis of keywords from a specific chapter is beneficial. Fig. 7 shows the keywords for chapters along with their associated scores derived from the XLNet+DDQL model.

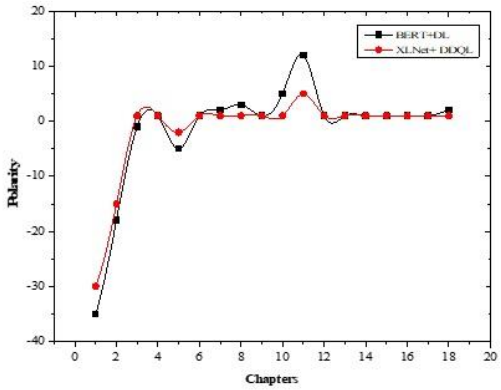


Fig. 6 Results of Arjuna's sentiments for chapters in Eknath Easwara translation

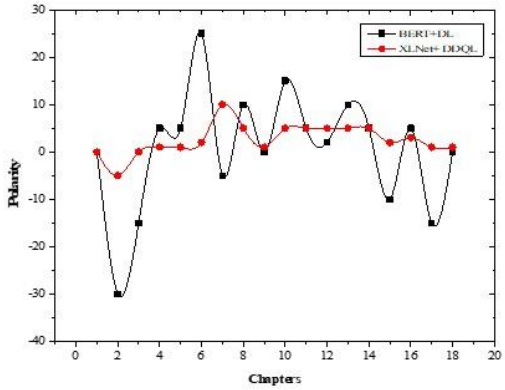


Fig. 7 Results of Lord Krishna's sentiments for chapters in Eknath Easwara translation

5. Conclusion

We explored sentiment analysis of Sanskrit texts using advanced deep learning methods. Our approach started by pre-processing the text using the NLP transformer to ensure accurate interpretation of the words given the word order. We then performed XLNet-based feature extraction to extract important features from the pre-processed text. To refine these features, we used a modified hyper spherical searching (MHSS) algorithm, which effectively reduces dimensionality by selecting optimal features. We used a dynamic two-layer Q-learning (DDQL) model for emotion classification and stress prediction. We validated the proposed XLNet+DDQL model by analyzing selected chapters and verses of Bhagavad Gita in different translations. The results show that our model significantly outperforms existing methods in translation and emotion classification tasks. In particular, the XLNet+DDQL model achieved an average Jaccard similarity score of 0.835 and a Dice similarity score of 0.825, representing a 12.234% improvement over the BERT model. Despite significant language and vocabulary variations, our sentiment analysis showed that the XLNet+DDQL model effectively identified and classified sentiment across chapters and translations.

References

1. Dash, B., Narasimham, G. L., &Jinde, P. (2024). Animals in Hinduism: Exploring Communication Beyond the Human Realm in Sacred Texts and Practices. Journal of Dharma Studies, 1-16.
2. Bonta, S. (2024). A semiotic grammar of Vedic Sanskrit. Chinese Semiotic Studies, 20(2), 379-423.
3. Srinivasan, R., &Aithal, P. S. (2024). Bhakti Blossoms: Tamil Poetry's Journey into Spiral Depths, Unraveling Devotion and Cultural Significance. International Journal of Philosophy and Languages (IJPL), 3(1), 23-35.
4. Fernandez, J. (2024). The Future of Religion Is Virtual Reality: Authenticating Religious Practice in the Coming Metaverse Through an Examination of Religion in Second Life (Master's thesis, California State University, Long Beach).
5. Surdeanu, M., & Valenzuela-Escárcega, M. A. (2024). Deep learning for natural language processing: a gentle introduction. Cambridge University Press.
6. Oraif, I. (2024). Natural Language Processing (NLP) and EFL Learning: A Case Study Based on Deep Learning. Journal of Language Teaching and Research, 15(1), 201-208.
7. Kaur, P., Kashyap, G. S., Kumar, A., Nafis, M. T., Kumar, S., &Shokeen, V. (2024). From Text to Transformation: A Comprehensive Review of Large Language Models' Versatility. arXiv preprint arXiv:2402.16142.
8. Rojas-Carabali, W., Agrawal, R., Gutierrez-Sinisterra, L., Baxter, S. L., Cifuentes-González, C., Wei, Y. C., ...&Agrawal, R. (2024). Natural Language Processing in Medicine and Ophthalmology: A Review for the 21st-century clinician. Asia-Pacific Journal of Ophthalmology, 100084.

9. Ashok, G., Ruthvik, N., &Jeyakumar, G. (2024, January). Optimizing Sentiment Analysis on Twitter: Leveraging Hybrid Deep Learning Models for Enhanced Efficiency. In International Conference on Distributed Computing and Intelligent Technology (pp. 179-192). Cham: Springer Nature Switzerland.
10. Akdemir, E., &Barışçı, N. (2024). A review on deep learning applications with semantics. Expert Systems with Applications, 124029.
11. RAJU, B., SRIRAM, D., & GANESAN, D. (2024). OPTIMIZED DEEP LEARNING MODEL FOR SENTIMENTAL ANALYSIS TO IMPROVE CONSUMER EXPERIENCE IN E-COMMERCE WEBSITES. Journal of Theoretical and Applied Information Technology, 102(7).
12. Rane, N., Choudhary, S., &Rane, J. (2024). Artificial intelligence, machine learning, and deep learning for sentiment analysis in business to enhance customer experience, loyalty, and satisfaction. Available at SSRN 4846145.
13. Dhikale, S. N. (2024). The Critical and comparative analysis of Ancient and Modern thoughts of Sanskrit Scholars on the 'Characteristics of Mahakavya'. Research Journal of Humanities and Social Sciences, 15, 2.
14. Nelakanti, A. K., Kulkarni, A., &Shailaj, N. (2024, February). START: Sanskrit Teaching; Annotation; and Research Tool–Bridging Tradition and Technology in Scholarly Exploration. In Proceedings of the 7th International Sanskrit Computational Linguistics Symposium (pp. 113-124).
15. Mukherjee, B., Majhi, D., Tiwari, P., &Chaudhary, S. (2024). Comparing Research Topics through Metatags Analysis: A Multi-module Machine Algorithm Approaches Using Real World Data on Digital Humanities. Journal of Scientometric Research, 13(1), 58-70.
16. Nigam, A., & Chandra, S. (2024, May). Dharmaśāstra Informatics: Concept Mining System for Socio-Cultural Facet in Ancient India. In Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation (pp. 30-39).
17. Kumar, S., &Vandanapu, M. K. (2024). Natural Language Generation and Artificial Intelligence in Financial Reporting: Transforming Financial Data into Strategic Insights for Executive Leadership. International Journal of Computer Engineering and Technology (IJCET), 15(2).
18. Lukwaro, E. A. E., Kalegele, K., &Nyambo, D. G. (2024). A Review on NLP Techniques and Associated Challenges in Extracting Features from Education Data. Int. J. Com. Dig. Sys, 16(1).
19. Mei, T., Zi, Y., Cheng, X., Gao, Z., Wang, Q., & Yang, H. (2024). Efficiency optimization of large-scale language models based on deep learning in natural language processing tasks. arXiv preprint arXiv:2405.11704.
20. Mohamed, Y. A., Khanan, A., Bashir, M., Mohamed, A. H. H., Adiel, M. A., &Elsadig, M. A. (2024). The impact of artificial intelligence on language translation: a review. Ieee Access, 12, 25553-25579.
21. Das, D., Guha, S., Brubaker, J. R., &Semaan, B. (2024, May). The``Colonial Impulse'' of Natural Language Processing: An Audit of Bengali Sentiment Analysis Tools and Their Identity-based Biases. In Proceedings of the CHI Conference on Human Factors in Computing Systems (pp. 1-18).
22. Kumar, P., Pathania, K., & Raman, B. (2023). Zero-shot learning based cross-lingual sentiment analysis for sanskrit text with insufficient labeled data. Applied Intelligence, 53(9), 10096-10113.
23. Khaparde, A., Deshmukh, V., &Kowdiki, M. (2023). Enhanced Nature-Inspired Algorithm-based Hybrid Deep Learning for Character Recognition in Sanskrit Language. Sensing and Imaging, 24(1), 23.
24. Qiao, R., & Huang, X. (2024). Application of Deep Learning in Cross-Lingual Sentiment Analysis for Natural Language Processing. Journal of Artificial Intelligence Practice, 7(1), 1-6.
25. Prakash, O., Kumar, S. V., Abbas, J. K., Naser, S. J., &Haroon, N. H. (2024, April). Enhancing Natural Language Processing with Deep Learning for Sentiment Analysis. In 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) (pp. 1-5). IEEE.
26. Tejaswini, V., SathyaBabu, K., &Sahoo, B. (2024). Depression detection from social media text analysis using natural language processing techniques and hybrid deep learning model. ACM Transactions on Asian and Low-Resource Language Information Processing, 23(1), 1-20.
27. Sruthi, S., A. (2024, March). Natural Language Processing for Sentiment Analysis with Deep Learning. In 2024 3rd International Conference for Innovation in Technology (INOCON) (pp. 1-6). IEEE.

28. Jain, D. K., Qamar, S., Sangwan, S. R., Ding, W., &Kulkarni, A. J. (2024). Ontology-Based Natural Language Processing for Sentimental Knowledge Analysis Using Deep Learning Architectures. ACM TALIP, 23(1), 1-17.
29. Molenaar, A., Lukose, D., Brennan, L., Jenkins, E. L., & McCaffrey, T. A. (2024). Using Natural Language Processing to Explore Social Media Opinions on Food Security: Sentiment Analysis and Topic Modeling Study. Journal of Medical Internet Research, 26, e47826.
30. Ahmed, W., Masood, T., Khushi, H. M. T., Akhter, S., Ghazanfar, M. N., &Naseer, I. (2024). Beyond Classification: Exploring the Potential of NLP and Deep Learning for Real-Time Sentiment Analysis on Twitter. Journal of Computing & Biomedical Informatics, 7(01), 157-166.
31. Chandra, R. and Kulkarni, V., 2022. Semantic and sentiment analysis of selected Bhagavad Gita translations using BERT-based language framework. IEEE Access, 10, pp.21291-21315.
32. Adam, E.E.B., 2020. Deep learning based NLP techniques in text to speech synthesis for communication recognition. Journal of Soft Computing Paradigm (JSCP), 2(04), pp.209-215.
33. Patil, R., Boit, S., Gudivada, V. and Nandigam, J., 2023. A survey of text representation and embedding techniques in nlp. IEEE Access, 11, pp.36120-36146.
34. Sharifani, K., Amini, M., Akbari, Y. and Aghajanzadeh Godarzi, J., 2022. Operating machine learning across natural language processing techniques for improvement of fabricated news model. International Journal of Science and Information System Research, 12(9), pp.20-44.
35. M. Maatuk, A. and A. Abdelnabi, E., 2021, April. Generating UML use case and activity diagrams using NLP techniques and heuristics rules. In International Conference on Data Science, E-learning and Information Systems 2021 (pp. 271-277).
36. Tusar, M.T.H.K. and Islam, M.T., 2021, September. A comparative study of sentiment analysis using NLP and different machine learning techniques on US airline Twitter data. In 2021 International Conference on Electronics, Communications and Information Technology (ICECIT) (pp. 1-4). IEEE.
37. Salloum, S., Gaber, T., Vadera, S. and Shaalan, K., 2021. Phishing email detection using natural language processing techniques: a literature survey. Procedia Computer Science, 189, pp.19-28.
38. Pan, J., 2024, April. Transductive sentiment analysis based on XlNet and GCN. , 5th International Conference on Computer Information and Big Data Applications (pp. 987-991).
39. Danyal, M.M., Khan, S.S., Khan, M, 2024. Proposing sentiment analysis model based on BERT and XLNet for movie reviews. Multimedia Tools and Applications, pp.1-25.
40. Wu, D., Wang, Z. and Zhao, W., 2024. XLNet-CNN-GRU dual-channel aspect-level review text sentiment classification method. Multimedia Tools and Applications, 83(2), pp.5871-5892.
41. Sathpathy, A.S., Mohanty, A., Ray, P.K., khan Bhutto, J., Alfiafi, A.J.M. and khulaif Alharbi, O., 2024. Hyper Spherical Search (HSS) algorithm based optimization and real-time stability study of Tidal energy conversion system. IEEE Access.
42. Deng, Y. and Xiang, X., 2024. Expanding hyperspherical space for few-shot class-incremental learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1967-1976).
43. Zhao, F., Zhuang, C., Wang, L. and Dong, C., 2024. An Iterative Greedy Algorithm With $ Q $-Learning Mechanism for the Multiobjective Distributed No-Idle Permutation Flowshop Scheduling. IEEE Transactions on Systems, Man, and Cybernetics: Systems.
44. Kamat, V.V., 2024. Bharat'kar and His Attempt Towards Saraswat Lusitanization 1. In The Colonial Periodical Press in the Indian and Pacific Ocean Regions (pp. 232-253). Routledge.
45. Chandra, R., Tiwari, A., Jain, N. and Badhe, S., 2024. Large language models for metaphor detection: Bhagavad Gita and Sermon on the Mount. IEEE Access.